Perform regression analysis on breast cancer

Report prepared by Xinyi Xie, Jiawei Zeng and Kexin Fan

December 9, 2022

Table of Contents

Background

Breast cancer is one of the most common cancers diagnosed in U.S. women. Many of us think of breast cancer as a female disease, but it can also occur in men. Breast cancer is found when the cells in the breast begin to grow uncontrollably. We filtered out some of the features in the progress of searching data. Based on these features, we can determine whether a tumor is benign(B) or malignant(M).

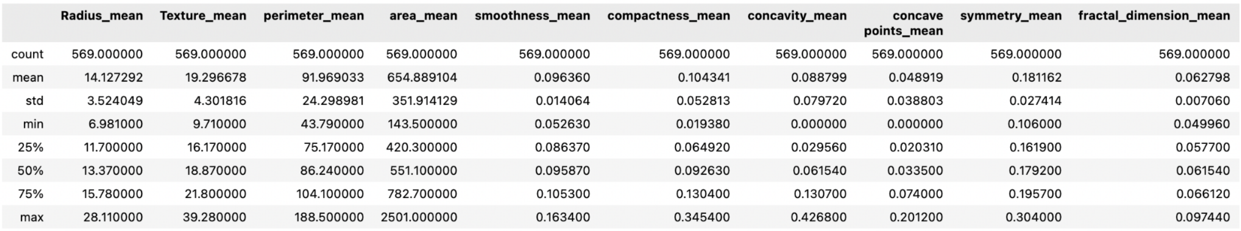
Objective

Our objective of this project is to explore which method is more effective in filtering out our data in many aspects.

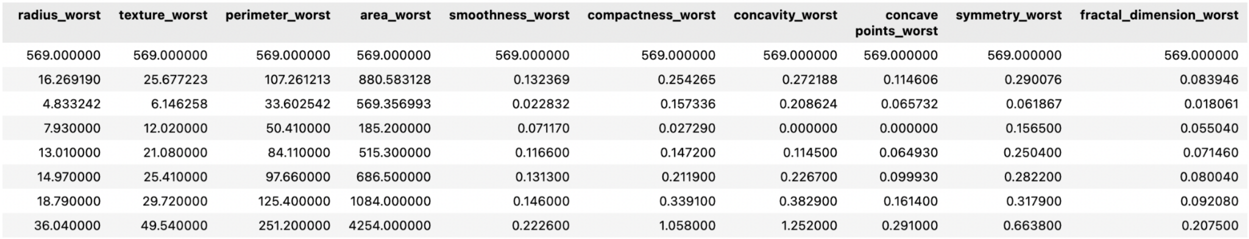
Dataset preparation

1. Clean and Generate data

We choose ID numbers and diagnosis as attribute factors and 10 real-valued features (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension) to predict. After we modifying dataframe and cleaning the missing value and outliers, we generate this continuous data:

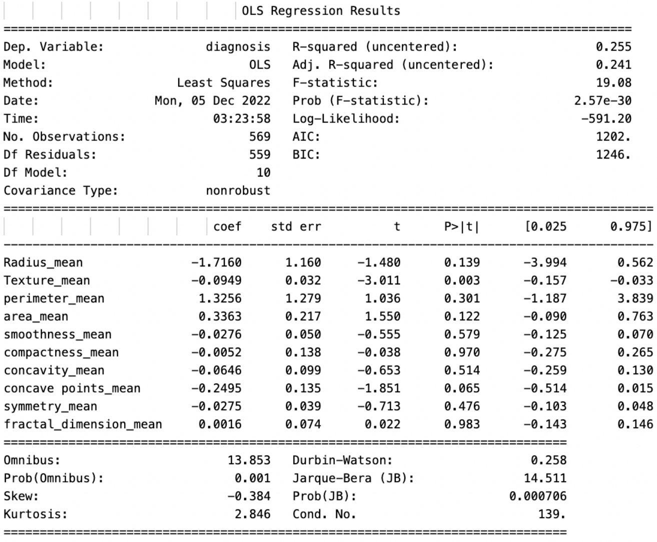


	Radius_mean	Texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.296678	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798
std	3.524049	4.301816	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700
50%	13.370000	18.870000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440



radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst
569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
16.269190	25.677223	107.261213	880.583128	0.132369	0.254265	0.272188	0.114606	0.290076	0.083946
4.833242	6.146258	33.602542	569.356993	0.022832	0.157336	0.208624	0.065732	0.061867	0.018061
7.930000	12.020000	50.410000	185.200000	0.071170	0.027290	0.000000	0.000000	0.156500	0.055040
13.010000	21.080000	84.110000	515.300000	0.116600	0.147200	0.114500	0.064930	0.250400	0.071460
14.970000	25.410000	97.660000	686.500000	0.131300	0.211900	0.226700	0.099930	0.282200	0.080040
18.790000	29.720000	125.400000	1084.000000	0.146000	0.339100	0.382900	0.161400	0.317900	0.092080
36.040000	49.540000	251.200000	4254.000000	0.222600	1.058000	1.252000	0.291000	0.663800	0.207500

We did an overall summary of data:

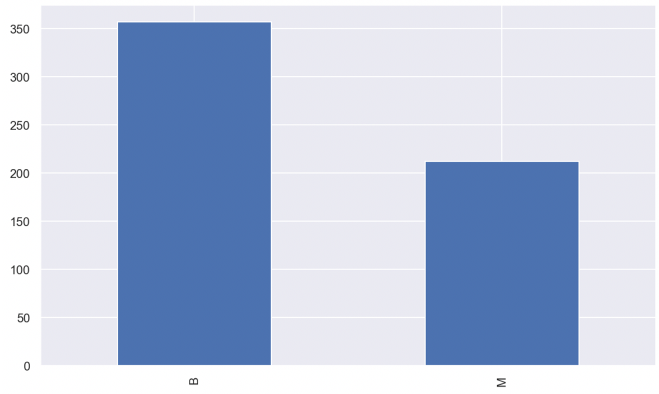


```
|   |   |   |   |   |   |   |   |OLS Regression Results
=======================================================================================
Dep. Variable:              diagnosis   R-squared (uncentered):                   0.255
Model:                            OLS   Adj. R-squared (uncentered):              0.241
Method:                 Least Squares   F-statistic:                              19.08
Date:                Mon, 05 Dec 2022   Prob (F-statistic):                    2.57e-30
Time:                        03:23:58   Log-Likelihood:                         -591.20
No. Observations:                 569   AIC:                                      1202.
Df Residuals:                     559   BIC:                                      1246.
Df Model:                          10
Covariance Type:            nonrobust
==========================================================================================
|   |   |   |   |   |   |   |   |coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Radius_mean              -1.7160      1.160     -1.480      0.139      -3.994       0.562
Texture_mean             -0.0949      0.032     -3.011      0.003      -0.157      -0.033
perimeter_mean            1.3256      1.279      1.036      0.301      -1.187       3.839
area_mean                 0.3363      0.217      1.550      0.122      -0.090       0.763
smoothness_mean          -0.0276      0.050     -0.555      0.579      -0.125       0.070
compactness_mean         -0.0052      0.138     -0.038      0.970      -0.275       0.265
concavity_mean           -0.0646      0.099     -0.653      0.514      -0.259       0.130
concave points_mean      -0.2495      0.135     -1.851      0.065      -0.514       0.015
symmetry_mean            -0.0275      0.039     -0.713      0.476      -0.103       0.048
fractal_dimension_mean    0.0016      0.074      0.022      0.983      -0.143       0.146
==============================================================================
Omnibus:                       13.853   Durbin-Watson:                   0.258
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               14.511
Skew:                          -0.384   Prob(JB):                     0.000706
Kurtosis:                       2.846   Cond. No.                         139.
==============================================================================
```

(补充

2. Barplot

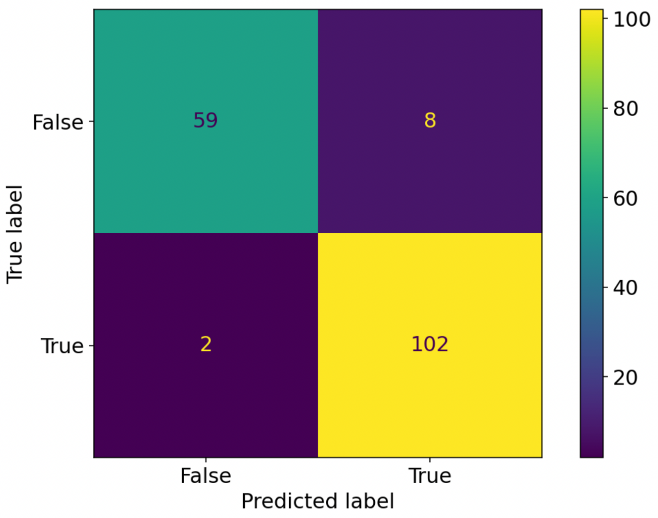
From the bar plot, we roughly estimate there is more benign (B) than malignant (M).



Method

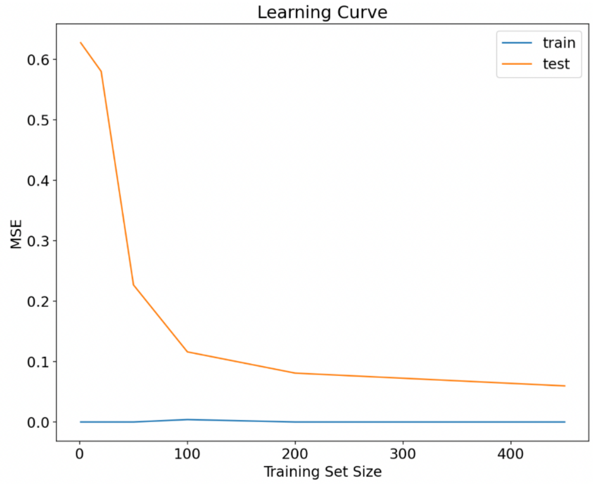
1. Random forest

a). Train and evaluate model



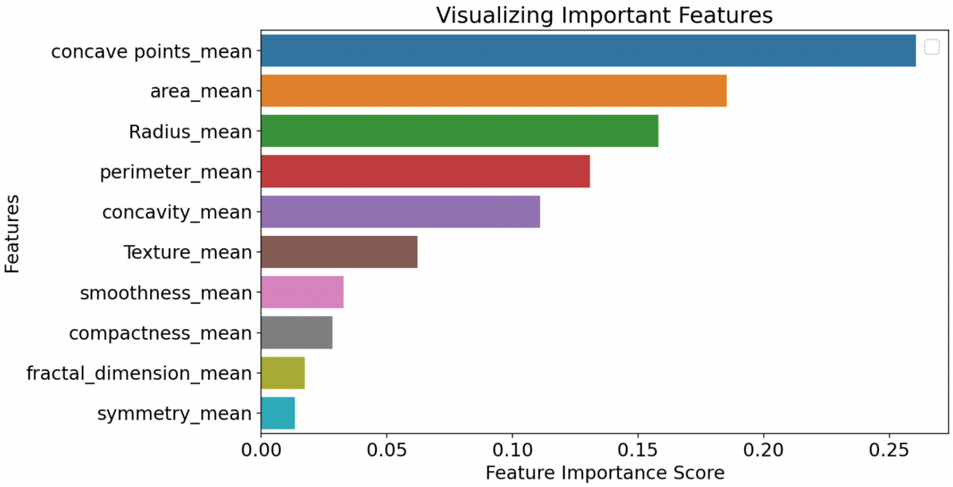
We set 30 trees to do the random forest. The accuracy score is 0.94152 which means we have 94.15% accurate, and the recall score is 0.98077 which is good as it is above 0.5. Then we calculate the variance and bias is 0.22947 and 0.4790327 respectively. Both of them are relatively low, which means that our model is not too bad.

b). Learning curve



The decision tree algorithm works 70% training set and 30% test set. From the learning curve, we found the accuracy rate is the highest when training set size at 100, MSE began to show a trend towards a steady decline around 0.1. Random 降varianceAda bias

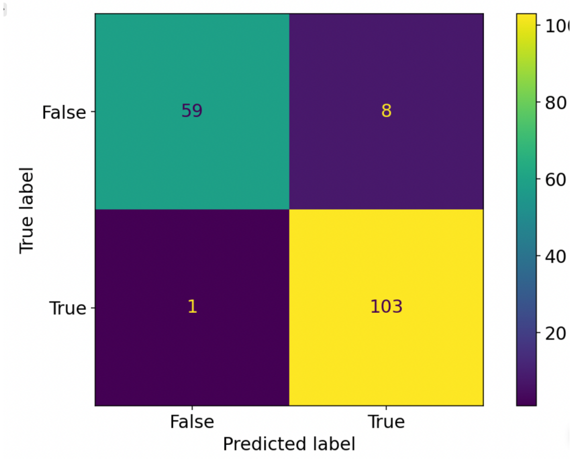
c). Feature selection



We did feature selection based on random forest. When we remove misleading data and noise, reduce some less important variables which are less than 0.05 , it approves our accuracy and also reduces the training time from 1.3s to 0.7s. This is more effective for our model than the original one.

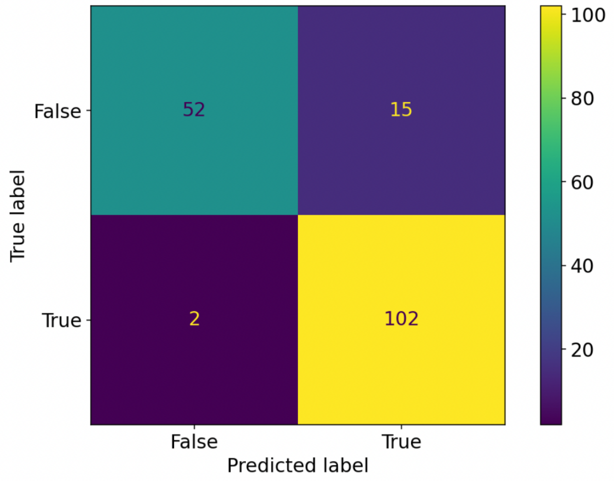
2. Ada boost

a). Train and evaluate



(based decision tree)

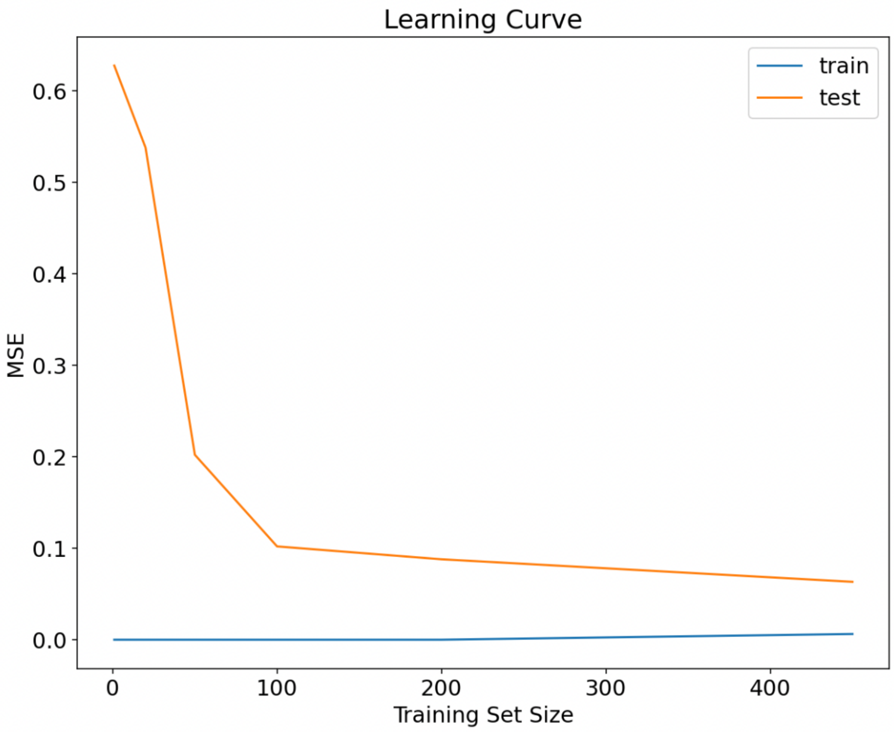
We use this method based on our original data. The accuracy score is 0.94737 which means we have 94.73% accuracy, higher than the accuracy score of random forest. And the recall score is 0.99034 which is a very good value. It shows that the ada boost is more effective than the random forest method.



(based support vector classifier)

The accuracy score is 90.058% and recall score is 0.98077. Comparing these two values with previous ones based on the decision tree, they are especially not as good as the previous one.

b). Learning curve



We got the same result as the random forest method, but better than random forest output, thus the learning curve obtained by adaboost is smoother and tends to be a more horizontal line at training set size equal to 100.

svc基于model会做很多遍计算，decisiontree还svc，svc侧重record score很高

Svc正常情况，全部的准确率不见得高，对能预测出正常情况的准确率是高的

Model comparison

Random forest and adaboost differ in the way samples are used. In random forest, the training data are sampled based on bagging, while in adaboost, it is based on boosting.

conclusion

At first we randomly set the number of estimated classifier trees to be 50, we generally believe that the number of trees is better when we have enough large size. Then secondly we tried to set the number of estimated classifier trees to be 30. After comparing the accuracy score and recall score under these two situations, we found that it is better when the number of trees is 30 because of the higher accuracy.

We speculated whether the smaller the number of trees set, the higher the accuracy. However, when we set the tree number to be 20, we found that the result of 30 is still better than 20. Therefore, we conclude that when we set the number of trees randomly, it should be setting appropriate number, not the larger the better.

Moreover, based on our previous result, we can conclude that the adaboost method is more effective than the random forest method. da+decision tree的时候最好。ada 方法也是decision